

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-313890

(43)Date of publication of application : 26.11.1993

(51)Int.Cl.

G06F 9/32

G06F 15/78

(21)Application number : 04-115562

(71)Applicant : NEC CORP

(22)Date of filing : 08.05.1992

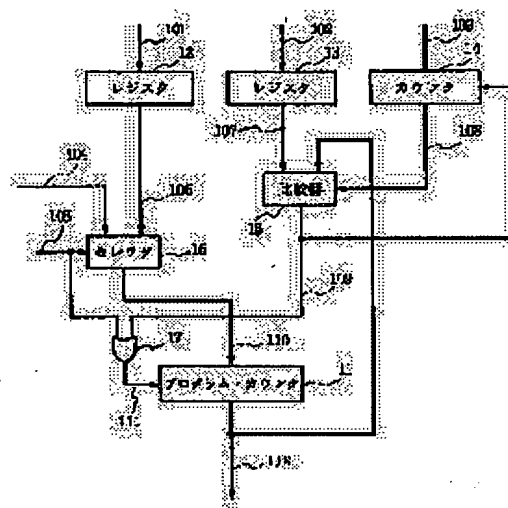
(72)Inventor : ISHIDA RYUJI

(54) ADDRESS ARITHMETIC CIRCUIT

(57)Abstract:

PURPOSE: To improve the deterioration of throughput due to a delay jump at the time of pipeline processing.

CONSTITUTION: The circuit is provided with a register 12 for inputting and storing a leading address signal 101 for loop processing, a register 13 for inputting and storing a final address signal 102, a counter 14 for inputting and holding a loop frequency data signal 103 and outputting a data signal 108 through a compared result signal 109, a selector 16 for inputting a jumping destination address signal 104 and a data signal 106 and selecting one of the signals 110 through a jump instruction signal 105 to be outputted, an OR circuit 17 for inputting the jump instruction signal 105 and the compared result signal 109 and outputting an OR, a program counter 11 for inputting a data signal 111 inputted from the OR circuit 17 and loading the data signal 110 selected and outputted at the selector 16 and a comparator 15 for inputting a data signal 112 outputted from the program counter 11 and a data signal 107 and outputting the compared result signal 109 through the data signal 108.



LEGAL STATUS

[Date of request for examination] 25.04.1996

[Date of sending the examiner's decision of rejection] 27.04.1999

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

*** NOTICES ***

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] In the address-arithmetic circuit for instruction memory read-out of a microprocessor The 1st data-hold means which inputs and stores the start-address signal for performing predetermined loop-formation processing, While inputting and holding the 2nd data-hold means which inputs and stores the last address signal for performing said loop-formation processing, and the loop count data signal corresponding to said loop-formation processing The 3rd data-hold means which outputs the level signal corresponding to the loop count data concerned through a predetermined comparison result signal, A predetermined jump place address signal and the data signal outputted from said 1st data-hold means are inputted. A data selection means to choose and output one signal of both the signals concerned through a predetermined jump indication signal, The OR operation circuit which inputs said jump indication signal and said comparison result signal, and generates and outputs the OR of both the signals concerned, The data signal outputted from said OR operation circuit is inputted as a timing signal for data incorporation. The program counter which is chosen in said data selection means and loads the data signal outputted through the timing signal concerned, While inputting the data signal outputted from said program counter, and the data signal outputted from said 2nd data storage means and carrying out comparison collating of the level of both the data signals concerned The address-arithmetic circuit characterized by having at least a data comparison means to output said comparison result signal through the data signal outputted from said 3rd data-hold means.

[Claim 2] The address-arithmetic circuit according to claim 1 characterized by controlling the output propriety of the comparison result signal outputted from said data comparison means by the level signal outputted from said 3rd data-hold means.

[Claim 3] [when it is shown in said data selection means that said jump indication signal is the timing which loads said jump place address signal to said program counter] [when it is shown that it is outside the timing to which the jump place address signal concerned is chosen, and it is outputted, and said jump indication signal loads said jump place address signal to said program counter] The address-arithmetic circuit according to claim 1 characterized by choosing and outputting the data signal outputted from said 1st data-hold means.

[Translation done.]

*** NOTICES ***

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.

2. **** shows the word which can not be translated.

3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Industrial Application] Especially this invention relates to the address-arithmetic circuit which functions as an object for read-out of instruction memory in a microprocessor about an address-arithmetic circuit.

[0002]

[Description of the Prior Art] Generally, a digital signal processor is mentioned as an example of the microprocessor which performs data processing at high speed. Moreover, a digital filter is mentioned as an application of a digital JIGUNARU processor.

[0003] In this kind of digital signal processor, in order to perform high-speed data processing, the technique of separating the memory area for instruction code storage, the address-arithmetic circuit for memory area directions, the data bus for instruction data transfer, and the memory area for operation data storage, the address-arithmetic circuit for memory area directions and the data bus for operation data transfer is used. As an address-arithmetic circuit of this memory for instruction code storage, the arithmetic circuit as shown below is used in the former:

[0004] The block diagram of an example of the conventional address-arithmetic circuit is shown in drawing 4. This conventional example is constituted by the program counter 41 which directs the address for instruction memory read-out, and is functioning as a rise counter which carries out the contents concerned +one whenever read-out of the contents of instruction memory currently held at the program counter 41 is performed. Usually, since a series of instruction codes are systematically recorded in the continuous memory area, processing is performed by reading said instruction code from the instruction memory of the address corresponding to the address value currently held in the program counter 41 by which a rise count is carried out.

[0005] Moreover, modification treatment of the effective procedure of an instruction is performed by using a JAMBU instruction-etc.; in case the flow of processing of this single string is changed in halfway, and transmitting this following effective address to a program counter 41, when this jump instruction is decoded by writing in beforehand the address of the instruction which should be executed on instruction code at the code and degree which show that it is jump instruction. Next, the address signal (jump place address signal) 125 of the instruction to execute is inputted, and the decoding signal (jump indication signal) 126 which shows that it is jump instruction is inputted. If the decoding signal 126 becomes active, in a program counter 41, an address signal 125 will be incorporated, a rise count will be repeatedly performed to the address value of this incorporated address signal 125, and, thereby, the flow of an instruction will be changed.

[0006] The timing chart showing the operating state in the case of making processing to read-out and an interpretation of instruction code into three steps of pipeline processing of read-out and decoding of address-arithmetic instruction memory is shown in drawing 5 (a), (b), (c), and (d), in drawing 5 (a), memory read-out processing and drawing 5 (c) show decoding, and, as for the address arithmetic of instruction memory, and drawing 5 (b), drawing 5 (d) shows the timing of data processing of operation.

[0007] Now, the m-th jump instruction shall be described by the n-th instruction address. in time

amount T, when the result of an operation by the address-arithmetic circuit is set to n, it comes out in time amount (T+6) that the instruction of Address n is actually executed, the time amount to which Address m is set to a program counter comes out, and there is in (T+3), and the instruction of Address m is executed. time amount (T+1) and time amount (T+2) — setting — the value of a program counter — with and (n+1) (n+2), it becomes, therefore the address (n+1) and (n+2) the instruction currently written in will be executed in time amount (T+4) and time amount (T+5), respectively. Describing the m-th jump instruction to the n-th street means that it is going to execute the instruction of the m-th henceforth as next processing in which the n-th street was performed, and, naturally (n+1), it needs to describe the instruction (NOP instruction) without anything at an address and (n+2) an address. Namely, the processing time while executing the above-mentioned NOP instruction (processing of what is not performed actually, either) turns into idle time amount, and will be in the condition of reducing the throughput as the whole.

[0008] Although this jump processing is used in case the multiple-times loop of the same instruction processing is carried out and it performs it especially, the fall of this throughput appears notably at the time of activation of the loop-formation instruction with which such jump processing is performed especially many times repeatedly. [many]

[0009]

[Problem(s) to be Solved by the Invention] In the conventional address-arithmetic circuit mentioned above In the case of the digital processor which meant improvement in the speed of processing and pipeline-processing-ized read-out processing of instruction memory etc. especially By delay of the processing which originates in this jump processing by the above-mentioned jump processing serving as a DIREIDO jump It becomes impossible to perform data processing of arbitration in the meantime, and there is a fault that the throughput of the data processing concerned declines remarkably, like a loop-formation instruction especially in data processing performed using much jump processings.

[0010]

[Means for Solving the Problem] The address-arithmetic circuit of this invention is set in the address-arithmetic circuit for instruction memory read-out of a microprocessor. The 1st data-hold means which inputs and stores the start-address signal for performing predetermined loop-formation processing, While inputting and holding the 2nd data-hold means which inputs and stores the last address signal for performing said loop-formation processing, and the loop count data signal corresponding to said loop-formation processing The 3rd data-hold means which outputs the level signal corresponding to the loop count data concerned through a predetermined comparison result signal, A predetermined jump place address signal and the data signal outputted from said 1st data-hold means are inputted. A data selection means to choose and output one signal of both the signals concerned through a predetermined jump indication signal, The OR operation circuit which inputs said jump indication signal and said comparison result signal, and generates and outputs the OR of both the signals concerned, The data signal outputted from said OR operation circuit is inputted as a timing signal for data incorporation. The program counter which is chosen in said data selection means and loads the data signal outputted through the timing signal concerned, While inputting the data signal outputted from said program counter, and the data signal outputted from said 2nd data storage means and carrying out comparison collating of the level of both the data signals concerned It has at least a data comparison means to output said comparison result signal through the data signal outputted from said 3rd data-hold means, and is constituted.

[0011] In addition, the output propriety of the comparison result signal outputted from said data comparison means may be constituted so that it may be controlled by the level signal outputted from

said 3rd data-hold means, and it sets for said data selection means. [when it is shown that said jump indication signal is the timing which loads said jump place address signal to said program counter] [when it is shown that it is outside the timing to which the jump place address signal concerned is chosen, and it is outputted, and said jump indication signal loads said jump place address signal to said program counter] It may be constituted so that the data signal outputted from said 1st data-hold means may be chosen and outputted.

[0012]

[Example] Next, this invention is explained with reference to a drawing.

[0013] Drawing 1 is the block diagram showing the 1st example of this invention. As shown in drawing 1, this example is equipped with a program counter 11, registers 12 and 13, a counter 14, a comparator 15, a selector 16, and OR circuit 17, and is constituted.

[0014] The contents of data are the rise counters carried out +one for every timing to which read-out of the contents of data by which a program counter 11 is held is performed, and the data signal 112 outputted from a program counter 11 is inputted into a comparator 15. Moreover, the data signal 110 outputted from a selector 16 is inputted into the program counter 11, and a data signal 110 is incorporated by the program counter 11 in the timing to which the level of the data signal 110 concerned becomes active. The data signal 107 which the data signal 106 which the start-address signal 101 and the last address signal 102 are inputted into registers 12 and 13, respectively, and is outputted from a register 12 is inputted into a selector 16, and is outputted from a register 13 is inputted into a comparator 15. The comparison result signal 109 outputted from the loop count status signal 103 and a comparator 15 is inputted into the counter 14, and the data signal 108 outputted from a counter 14 is inputted into a comparator 15. As a data input which shows the timing of the count in a counter 14, the comparison result signal 109 outputted from the above-mentioned comparator 15 corresponds, and the contents of data currently held in the counter 14 are carried out -one in the timing to which this comparison result signal 109 becomes active.

[0015] As an input to a comparator 15, as mentioned above, there are data signals 107, 112, and 108 outputted from a register 13, a program counter 11, and a counter 14, respectively, and the comparison result signal 109 outputted from a comparator 15 is inputted into a counter 14 and OR circuit 17. In this case, when the data signal 108 inputted into a comparator 15 from a counter 14 is "H" level, the output of the comparison result signal 109 outputted from a comparator 15 becomes good, and when a data signal 108 is "L" level, the output of the comparison result signal 109 is forbidden. Moreover, to the selector 16, the data signal 106 outputted from the jump place address 104, the jump indication signal 105, and a register 12 is inputted, and the data signal 110 outputted from a selector 16 is inputted into a program counter 11. To OR circuit 17, the comparison result signal 109 outputted from the above-mentioned jump indication signal 105 and an above-mentioned comparator 15 is inputted, and the OR output signal 111 of both these signals is inputted into a program counter 11.

[0016] About procedure until it results in read-out and an interpretation of the instruction memory in this example, it shall be based on an address arithmetic, memory read-out, and the three-step pipeline processing of decoding like the case of the above-mentioned conventional example. The timing of operation at the time of the usual jump actuation is the same as that of the case of the conventional example, and comes to be shown in drawing 2 (a), (b), (c), and (d). As mentioned above, in drawing 2 (a), memory read-out processing and drawing 2 (c) show decoding, and, as for the address arithmetic of instruction memory, and drawing 2 (b), drawing 2 (d) shows the timing of data processing of operation.

[0017] Now, let loop count be gamma time by making the last address into beta address, making as alpha address the start address to which loop-formation processing is performed. In the conventional loop-

formation processing, the loop-formation processing concerned is realized by describing a conditional-jump instruction at alpha address at the address which is the next address of beta address of the last address with which loop-formation processing is performed (beta+1). In case instruction description is mechanically changed so that I may be understood by this, the start address, the last address, and loop count of loop-formation processing have become clear in advance. For this reason, this loop-formation instruction is described by the address (alpha-1) in this invention. The data which show the start address which performs a loop-formation operation, the last address, and loop count to this loop-formation instruction are described. In addition, about a start address, it is also possible to hold as instruction code by interpreting it as it being the next instruction which described the loop-formation instruction, and also to ask from a current program counter value using an arithmetic circuit. for example, in time amount t, when a program counter value is set to (beta-1), in time amount (t+3), it comes out that a loop-formation instruction is decoded and executed, there is, and the value of the program counter at this time is set to (alpha+1). That is, (the present program counter value -2), it can ask for a start address by calculating.

[0018] If a loop-formation instruction is executed, the start-address signal 101 will be inputted and set as a register 12, and the last address signal 102 will be inputted and set as a register 13. Moreover, the loop count status signal 103 is inputted into a counter 14, and is set as coincidence. In here, since the contents of data currently held at the counter 14 are larger than "L" level, the data signal 108 has become active, and the comparison result signal 109 outputted from a comparator 15 by this becomes effective, and is inputted into a counter 14 and OR circuit 17. When in agreement with the value with which the value of a program counter 11 is held at the register 13, i.e., the loop-formation last address, the comparison result signal 109 is outputted as an active coincidence signal. Thereby, the OR output signal 111 outputted from OR circuit 17 also becomes active, and, in response, the data signal 110 outputted from a selector 16 is incorporated in a program counter 11. since [in this case,] jump processing is not performed — the jump indication signal 105 — non, it is in the active condition, the latter data signal 106 of the jump place address signal 104 and the data signals 106 outputted from a register 12 is chosen in a selector 16 by this, and it is outputted as a data signal 110. Therefore, the address data incorporated at this time are start-address alpha currently held at the register 12.

Moreover, since the comparison result signal 109 inputted into coincidence from a comparator 15 in a counter 14 is AKUREIBU, a down count is performed, and the data currently held are carried out -one. In this way, the data currently held at the counter 14 serve as "L" level by performing loop-formation processing of the count of a convention. Thereby, comparator 15 output will be in the condition of an invalid.

[0019] Next, since start-address alpha currently held at the register 12 is not set up in a program counter 11 when the value of an address counter 11 is in agreement with last address beta, subsequent instructions are executed succeeding (beta+1). In time amount T, the value of the program counter 11 at the time of the value of a program counter 11 being in agreement with last address beta and the timing of the circumference of instruction memory of operation are shown in drawing 2 (a), (b), (c), and (d). This drawing 2 (a), (b), (c), and (d) so that I may be understood by carrying out comparison collating with above-mentioned drawing 5 (a), (b), (c), and (d) In one-time loop-formation processing, it turns out in this invention that reduction of the processing time of 3 instruction cycle is attained including the jump processing described by the n-th street in drawing 5 (a), (b), (c), and (d). For example, when loop-formation processing of five instructions is performed 64 times, in the conventional configuration, the execution time of the instruction cycle of $x(5+3)64=512$ is needed [including activation of the first loop-formation instruction] in this invention to activation being completed by $1+5 \times 64=321$ instruction cycle.

[0020] In addition, about the usual jump processing, in the case of this example, the jump place address signal 104 is inputted into a selector 16, and the jump indication signal 105 is inputted into a selector 16 and OR circuit 17. In a selector 16, through the jump indication signal 105, it is chosen and outputted, and the jump place address signal 104 is inputted into a program counter 11 as a data signal 110, and is incorporated at this time. When the jump indication signal 105 becomes effective at this time, it is also possible to force loop-formation processing to terminate by adding separately the circuit which clears the value of a counter 14.

[0021] Next, the 2nd example of this invention is explained. Drawing 3 is the block Fig. showing the 2nd example concerned. As shown in drawing 3, this example is equipped with a program counter 31, registers 32 and 33, a flag 34, a comparator 35, a selector 36, and OR circuit 37, and is constituted.

[0022] The difference with the 1st example of this example is that the counter 14 shown in drawing 1 as one of the components in the 1st example is transposed to the flag 34 in this example. By this, the level of the data signal 120 which judges the output propriety of the comparison result signal 121 in a comparator 35 will be based on the data currently held at the flag 34. Although it is necessary during a loop-formation operation to operate this flag 34 by this, it becomes possible by operating this flag 34 to perform processing in which it extracts from the loop-formation processing concerned, according to the specific conditions under loop-formation processing. About other actuation, it is completely the same as that of the case of the 1st above-mentioned example.

[0023]

[Effect of the Invention] As explained above, this invention sets up beforehand the start address and the last address of loop-formation processing before the loop-formation processing initiation concerned, and carries out comparison collating of the value of a program counter, and the value of said last address. With the effectiveness of becoming possible to reduce the time delays by implementation of jump instruction by having the circuit which sets a start address as a program counter when both are in agreement Activation of the jump instruction itself also becomes unnecessary and is effective in the ability to reduce the number of execute steps of loop-formation processing, and improve processing efficiency.

[Translation done.]

* NOTICES *

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the block diagram showing the 1st example of this invention.

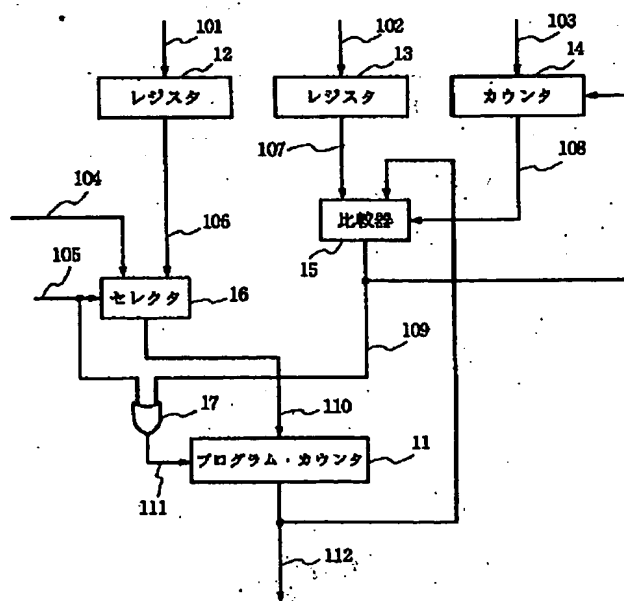
[Drawing 2] It is a timing chart of operation in the 1st example.

(11)特許出願公開番号

(43)公開日 平成5年(1993)11月26日

510 A 7530-5L

(74)代理人 弁理士 京本 直樹 (外2名)



(2)

【特許請求の範囲】

【請求項1】 マイクロプロセッサの命令メモリ読出し用のアドレス演算回路において、
 所定のループ処理を行うための先頭アドレス信号を入力して格納する第1のデータ保持手段と、
 前記ループ処理を行うための最終アドレス信号を入力して格納する第2のデータ保持手段と、
 前記ループ処理に対応するループ回数データ信号を入力して保持するとともに、所定の比較結果信号を介して、
 当該ループ回数データに対応するレベル信号を出力する第3のデータ保持手段と、
 所定の飛び先アドレス信号と、前記第1のデータ保持手段より出力されるデータ信号とを入力して、所定のジャンプ指示信号を介して当該両信号の内の一方の信号を選択して出力するデータ選択手段と、
 前記ジャンプ指示信号と前記比較結果信号とを入力して、当該両信号の論理和を生成して出力する論理和演算回路と、
 前記論理和演算回路より出力されるデータ信号をデータ取込み用のタイミング信号として入力し、前記データ選択手段において選択されて出力されるデータ信号を、当該タイミング信号を介してロードするプログラム・カウンタと、
 前記プログラム・カウンタより出力されるデータ信号と、前記第2のデータ格納手段より出力されるデータ信号とを入力して、当該両データ信号のレベルを比較照合するとともに、前記第3のデータ保持手段より出力されるデータ信号を介して前記比較結果信号を出力するデータ比較手段と、
 を少なくとも備えることを特徴とするアドレス演算回路。

【請求項2】 前記データ比較手段より出力される比較結果信号の出力可否が、前記第3のデータ保持手段より出力されるレベル信号により制御されることを特徴とする請求項1記載のアドレス演算回路。

【請求項3】 前記データ選択手段において、前記ジャンプ指示信号が、前記飛び先アドレス信号を前記プログラム・カウンタにロードするタイミングであることを示す時点において、当該飛び先アドレス信号が選択されて出力され、また、前記ジャンプ指示信号が、前記飛び先アドレス信号を前記プログラム・カウンタにロードするタイミング外であることを示す時点において、前記第1のデータ保持手段より出力されるデータ信号が選択されて出力されることを特徴とする請求項1記載のアドレス演算回路。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明はアドレス演算回路に関し、特にマイクロプロセッサにおいて、命令メモリの読出し用として機能するアドレス演算回路に関する。

【0002】

【従来の技術】 一般に、高速にて演算処理を行うマイクロプロセッサの例としては、ディジタル・シグナル・プロセッサが挙げられる。また、ディジタル・シグナル・プロセッサの応用例としては、ディジタル・フィルタが挙げられる。

【0003】 この種のディジタル・シグナル・プロセッサにおいては、高速演算処理を実行するために、命令コード記憶用のメモリ領域、メモリ領域指示用アドレス演算回路、命令データ転送用データバスと、演算データ記憶用メモリ領域、メモリ領域指示用アドレス演算回路、演算データ転送用データバスとを分離する手法が用いられている。この命令コード記憶用メモリのアドレス演算回路としては、従来においては、以下に示されるような演算回路が用いられている。

【0004】 図4に示されるのは、従来のアドレス演算回路の一例のブロック図である。本従来例は、命令メモリ読出し用のアドレスを指示するプログラムカウンタ41により構成されており、プログラムカウンタ41に保持されている命令メモリ内容の読出しが行われるたびに、当該内容を+1してゆくアップカウンタとして機能している。通常は、一連の命令コードが、連続するメモリ領域内において順序立てて記録されているため、アップカウントされてゆくプログラムカウンタ41内に保持されているアドレス値に対応するアドレスの命令メモリから前記命令コードを読出すことにより処理が実行される。

【0005】 また、この一連の処理の流れを中途において変更する際には、ジャンプ命令等が用いられて、命令コード上にジャンプ命令であることを示すコードおよび次に実行すべき命令のアドレスを予め書込んでおくことにより、このジャンプ命令をデコードした際に、この次の実行アドレスをプログラムカウンタ41に転送することにより、命令の実効手順の変更処理が行われる。次に、実行する命令のアドレス信号（とび先アドレス信号）125が入力されて、ジャンプ命令であることを示すデコード信号（ジャンプ指示信号）126が入力される。デコード信号126がアクティブになると、プログラムカウンタ41においては、アドレス信号125が取込まれて、この取込まれたアドレス信号125のアドレス値に対してアップカウントが繰返して行われ、これにより命令の流れが変更される。

【0006】 図5（a）、（b）、（c）および（d）に示されるのは、命令コードの読出し・解釈までの処理を、アドレス演算命令メモリの読出し・デコード処理の3段のパイプライン処理とする場合の動作状態を示すタイミング図であり、図5（a）は命令メモリのアドレス演算、図5（b）はメモリ読出し処理、図5（c）はデコード処理、そして図5（d）は演算処理の動作タイミングを示している。

(3)

3

【0007】今、命令アドレス n 番地に、 m 番地へのジャンプ命令が記述されているものとする。時間 T において、アドレス演算回路による演算結果が n となった場合、実際にアドレス n の命令が実行されて、プログラム・カウンタに対してアドレス m が設定される時間は、 $(T+3)$ においてであり、アドレス m の命令が実行されるのは、時間 $(T+6)$ においてである。時間 $(T+1)$ および時間 $(T+2)$ においては、プログラム・カウンタの値は $(n+1)$ および $(n+2)$ となり、従って、時間 $(T+4)$ および時間 $(T+5)$ においては、それぞれアドレス $(n+1)$ および $(n+2)$ に書込まれていた命令が実行されることになる。 n 番地に m 番地へのジャンプ命令を記述するということは、 n 番地を実行した次の処理として、 m 番地以降の命令を実行しようとすることを意味しており、当然 $(n+1)$ 番地および $(n+2)$ 番地には、何もない命令(NOP命令)を記述する必要がある。即ち、上記のNOP命令を実行(現実には何の処理も行われない)している間の処理時間が無為の時間となり、全体としての処理能力を低下させる状態となる。

【0008】このジャンプ処理は、特に同一の命令処理を複数回繰返して実行する際に多く用いられるが、特に、このようなジャンプ処理が多数回繰返して行われるループ命令の実行時においては、この処理能力の低下が顕著に表われる。

【0009】

【発明が解決しようとする課題】上述した従来のアドレス演算回路においては、特に処理の高速化を意図して、命令メモリの読出し処理等をパイプライン処理化したデジタル・プロセッサの場合には、前述のジャンプ処理がディレイド・ジャンプとなり、このジャンプ処理に起因する処理の遅延により、この間において任意の演算処理を実行することが不可能となり、特に、ループ命令のように、ジャンプ処理を多数用いて行われる演算処理においては、当該演算処理の処理能力が著しく低下するという欠点がある。

【0010】

【課題を解決するための手段】本発明のアドレス演算回路は、マイクロプロセッサの命令メモリ読出し用のアドレス演算回路において、所定のループ処理を行うための先頭アドレス信号を入力して格納する第1のデータ保持手段と、前記ループ処理を行うための最終アドレス信号を入力して格納する第2のデータ保持手段と、前記ループ処理に対応するループ回数データ信号を入力して保持するとともに、所定の比較結果信号を介して、当該ループ回数データに対応するレベル信号を出力する第3のデータ保持手段と、所定の飛び先アドレス信号と、前記第1のデータ保持手段より出力されるデータ信号とを入力して、所定のジャンプ指示信号を介して当該両信号の内、一方の信号を選択して出力するデータ選択手段と、前

4

記ジャンプ指示信号と前記比較結果信号とを入力して、当該両信号の論理和を生成して出力する論理和演算回路と、前記論理和演算回路より出力されるデータ信号をデータ取込み用のタイミング信号として入力し、前記データ選択手段において選択されて出力されるデータ信号を、当該タイミング信号を介してロードするプログラム・カウンタと、前記プログラム・カウンタより出力されるデータ信号と、前記第2のデータ格納手段より出力されるデータ信号とを入力して、当該両データ信号のレベルを比較照合するとともに、前記第3のデータ保持手段より出力されるデータ信号を介して前記比較結果信号を出力するデータ比較手段と、を少なくとも備えて構成される。

【0011】なお、前記データ比較手段より出力される比較結果信号の出力可否が、前記第3のデータ保持手段より出力されるレベル信号により制御されるように構成されてもよく、また、前記データ選択手段において、前記ジャンプ指示信号が、前記飛び先アドレス信号を前記プログラム・カウンタにロードするタイミングであることを示す時点において、当該飛び先アドレス信号が選択されて出力され、また、前記ジャンプ指示信号が、前記飛び先アドレス信号を前記プログラム・カウンタにロードするタイミング外であることを示す時点において、前記第1のデータ保持手段より出力されるデータ信号が選択されて出力されるように構成されてもよい。

【0012】

【実施例】次に、本発明について図面を参照して説明する。

【0013】図1は本発明の第1の実施例を示すブロック図である。図1に示されるように、本実施例は、プログラム・カウンタ11と、レジスタ12および13と、カウンタ14と、比較器15と、セクタ16と、OR回路17とを備えて構成される。

【0014】プログラム・カウンタ11は、保持されるデータ内容の読出しが行われるタイミングごとに、そのデータ内容が+1されるアップカウンタであり、プログラム・カウンタ11より出力されるデータ信号112は比較器15に入力される。また、プログラム・カウンタ11には、セクタ16より出力されるデータ信号110が入力されており、当該データ信号110のレベルがアクティブになるタイミングにおいて、データ信号110はプログラム・カウンタ11に取込まれる。レジスタ12および13には、それぞれ先頭アドレス信号101および最終アドレス信号102が入力されており、レジスタ12より出力されるデータ信号106は、セクタ16に入力され、またレジスタ13より出力されるデータ信号107は比較器15に入力される。カウンタ14にはループ回数表示信号103および比較器15より出力される比較結果信号109が入力されており、カウンタ14より出力されるデータ信号108は、比較器15

(4)

5

に入力される。カウンタ14におけるカウントのタイミングを示すデータ入力としては、上述の比較器15より出力される比較結果信号109が対応しており、この比較結果信号109がアクティブになるタイミングにおいて、カウンタ14において保持されているデータ内容は-1される。

【0015】比較器15に対する入力としては、前述のように、レジスタ13、プログラム・カウンタ11およびカウンタ14よりそれぞれ出力されるデータ信号107、112および108があり、比較器15より出力される比較結果信号109は、カウンタ14およびOR回路17に入力される。この場合、カウンタ14より比較器15に入力されるデータ信号108が“H”レベルの時にのみ、比較器15より出力される比較結果信号109の出力が可となり、データ信号108が“L”レベルの時には、比較結果信号109の出力は禁止される。また、セクタ16に対しては、飛び先アドレス104、ジャンプ指示信号105およびレジスタ12より出力されるデータ信号106が入力されており、セクタ16より出力されるデータ信号110はプログラム・カウンタ11に入力される。OR回路17に対しては、上述のジャンプ指示信号105および比較器15より出力される比較結果信号109が入力されており、これらの両信号の論理和出力信号111は、プログラム・カウンタ11に入力される。

【0016】本実施例における命令メモリの読出し・解釈に至るまでの手続きについては、前述の従来例の場合と同様に、アドレス演算、メモリ読出しおよびデコードの3段パイプライン処理によるものとする。通常のジャンプ動作時における動作タイミングは、従来例の場合と同様で、図2(a)、(b)、(c)および(d)に示されるようになる。前述のように、図2(a)は命令メモリのアドレス演算、図2(b)はメモリ読出し処理、図2(c)はデコード処理、そして図2(d)は演算処理の動作タイミングを示している。

【0017】今、ループ処理が行われる先頭アドレスを α 番地、最終アドレスを β 番地として、ループ回数を γ 回とする。従来のループ処理においては、ループ処理が行われる最終番地の β 番地の次の番地である $(\beta+1)$ 番地に、 α 番地への条件付ジャンプ命令を記述することにより、当該ループ処理が実現されている。これにより理解されるように、命令記述を機械的に変換する際には、ループ処理の先頭アドレス、最終アドレスおよびループ回数は事前に判明している。このために、本発明においては、このループ命令は $(\alpha-1)$ 番地に記述される。このループ命令には、ループ演算を行う先頭アドレス、最終アドレスおよびループ回数を示すデータが記述されている。なお、先頭アドレスについては、ループ命令を記述した次の命令であると解釈することにより、命令コードとして保持する他に、演算回路を用いて現在の

6

プログラム・カウンタ値より求めることも可能である。例えば、時間 t において、プログラム・カウンタ値が $(\beta-1)$ となった場合には、ループ命令がデコードされて実行されるのは時間 $(t+3)$ においてであり、この時のプログラム・カウンタの値は $(\alpha+1)$ となる。即ち(現在のプログラム・カウンタ値-2)の演算を行うことにより、先頭アドレスを求めることができる。

【0018】ループ命令が実行されると、先頭アドレス信号101はレジスタ12に入力されて設定され、また最終アドレス信号102はレジスタ13に入力されて設定される。また、同時に、ループ回数表示信号103はカウンタ14に入力されて設定される。ここにおいて、カウンタ14に保持されているデータ内容が“L”レベルよりも大きいために、データ信号108はアクティブとなっており、これにより比較器15より出力される比較結果信号109は有効となり、カウンタ14およびOR回路17に入力される。プログラム・カウンタ11の値がレジスタ13に保持されている値、即ちループ最終アドレスと一致する場合には、比較結果信号109はアクティブな一致信号として出力される。これにより、OR回路17より出力される論理和出力信号111もアクティブとなり、これを受けて、プログラム・カウンタ11においては、セクタ16より出力されるデータ信号110が取込まれる。この際にはジャンプ処理が行われていないために、ジャンプ指示信号105はノンアクティブの状態となっており、これにより、セクタ16においては、飛び先アドレス信号104とレジスタ12より出力されるデータ信号106の内の、後者のデータ信号106が選択されて、データ信号110として出力される。従って、この時点において取込まれるアドレス・データは、レジスタ12に保持されている先頭アドレス α である。また、同時に、カウンタ14においては、比較器15より入力される比較結果信号109がアクティブであるためにダウン・カウントが行われ、保持されているデータは-1される。かくして、規定回数のループ処理が実行されることにより、カウンタ14に保持されているデータは“L”レベルとなる。これにより、比較器15出力は無効の状態となる。

【0019】次に、アドレス・カウンタ11の値が最終アドレス β と一致する場合には、プログラム・カウンタ11においては、レジスタ12に保持されている先頭アドレス α が設定されることがないために、引続き $(\beta+1)$ 以降の命令が実行されてゆく。図2(a)、(b)、(c)および(d)には、時間Tにおいて、プログラム・カウンタ11の値が最終アドレス β に一致した際におけるプログラム・カウンタ11の値および命令メモリまわりの動作タイミングが示される。この図2(a)、(b)、(c)および(d)を、前述の図5(a)、(b)、(c)および(d)と比較照合することにより理解されるように、一度のループ処理におい

(5)

て、図5(a)、(b)、(c)および(d)におけるn番地に記述されているジャンプ処理を含めて、本発明においては、3命令サイクルの処理時間の低減が可能となることが分かる。例えば、5命令のループ処理を64回実行した場合、本発明においては、最初のループ命令の実行を含めて、 $1 + 5 \times 64 = 321$ 命令サイクルで実行が終了するのに対して、従来の構成においては、

$(5 + 3) \times 64 = 512$ の命令サイクルの実行時間が必要となる。

【0020】なお、通常のジャンプ処理については、本実施例の場合には、飛び先アドレス信号104がセクタ16に入力され、またジャンプ指示信号105がセクタ16およびOR回路17に入力される。この時、セクタ16においては、ジャンプ指示信号105を介して飛び先アドレス信号104が選択されて出力され、データ信号110としてプログラム・カウンタ11に入力されて取込まれる。この時に、ジャンプ指示信号105が有効となる場合には、カウンタ14の値をクリアする回路を別途付加することにより、ループ処理の強制終了を行うことも可能である。

【0021】次に、本発明の第2の実施例について説明する。図3は、当該第2の実施例を示すブロック図である。図3に示されるように、本実施例は、プログラム・カウンタ31と、レジスタ32および33と、フラグ34と、比較器35と、セクタ36と、OR回路37とを備えて構成される。

【0022】本実施例の第1の実施例との相違点は、第1の実施例における構成要素の一つとして図1に示されているカウンタ14が、本実施例においてはフラグ34に置換えられていることである。これにより、比較器35における比較結果信号121の出力可否を判定するデータ信号120のレベルが、フラグ34に保持されているデータによることになる。これにより、ループ演算中

において、このフラグ34を操作することが必要となるが、しかし、このフラグ34を操作することにより、ループ処理中の特定条件により、当該ループ処理から抜出すという処理を行うことが可能となる。その他の動作については、前述の第1の実施例の場合と全く同様である。

【0023】

【発明の効果】以上説明したように、本発明は、ループ処理の先頭アドレスと最終アドレスとを当該ループ処理開始前において予め設定しておき、プログラム・カウンタの値と前記最終アドレスの値とを比較照合して、両者が一致する場合に先頭アドレスをプログラム・カウンタに設定する回路を備えることにより、ジャンプ命令の実施による遅延時間を削減することが可能になるという効果とともに、ジャンプ命令自体の実行も不要となり、ループ処理の実行ステップ数を低減して処理能率を改善することができるという効果がある。

【図面の簡単な説明】

【図1】本発明の第1の実施例を示すブロック図である。

【図2】第1の実施例における動作タイミング図である。

【図3】本発明の第2の実施例を示すブロック図である。

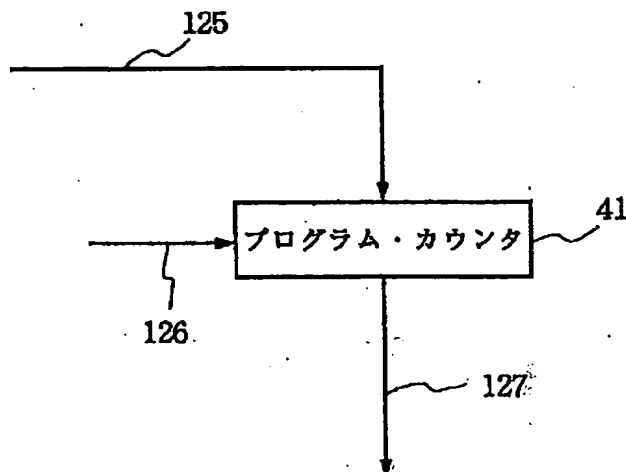
【図4】従来例を示すブロック図である。

【図5】従来例における動作タイミング図である。

【符号の説明】

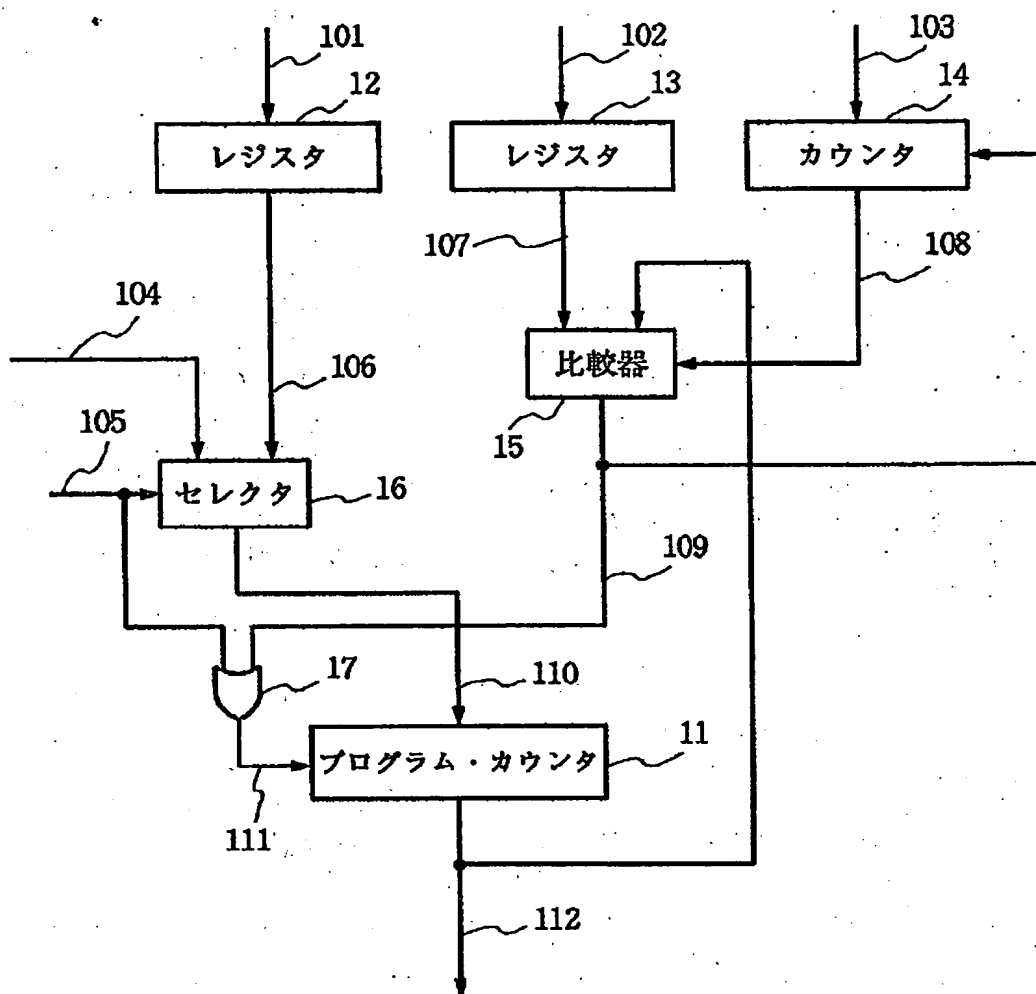
11、31、41	プログラム・カウンタ
12、13、32、33	レジスタ
14	カウンタ
15、35	比較器
16、36	セクタ
17、37	OR回路

【図4】



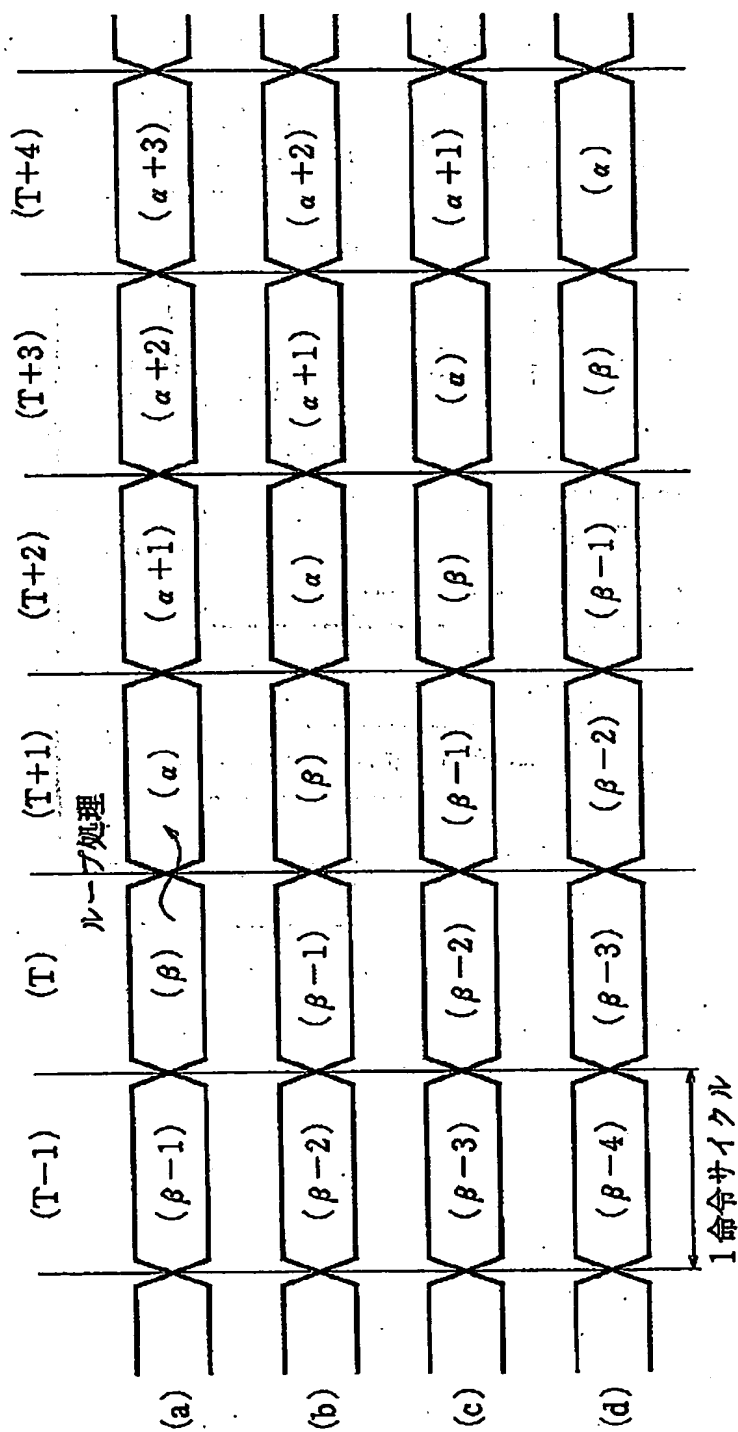
(6)

【図1】



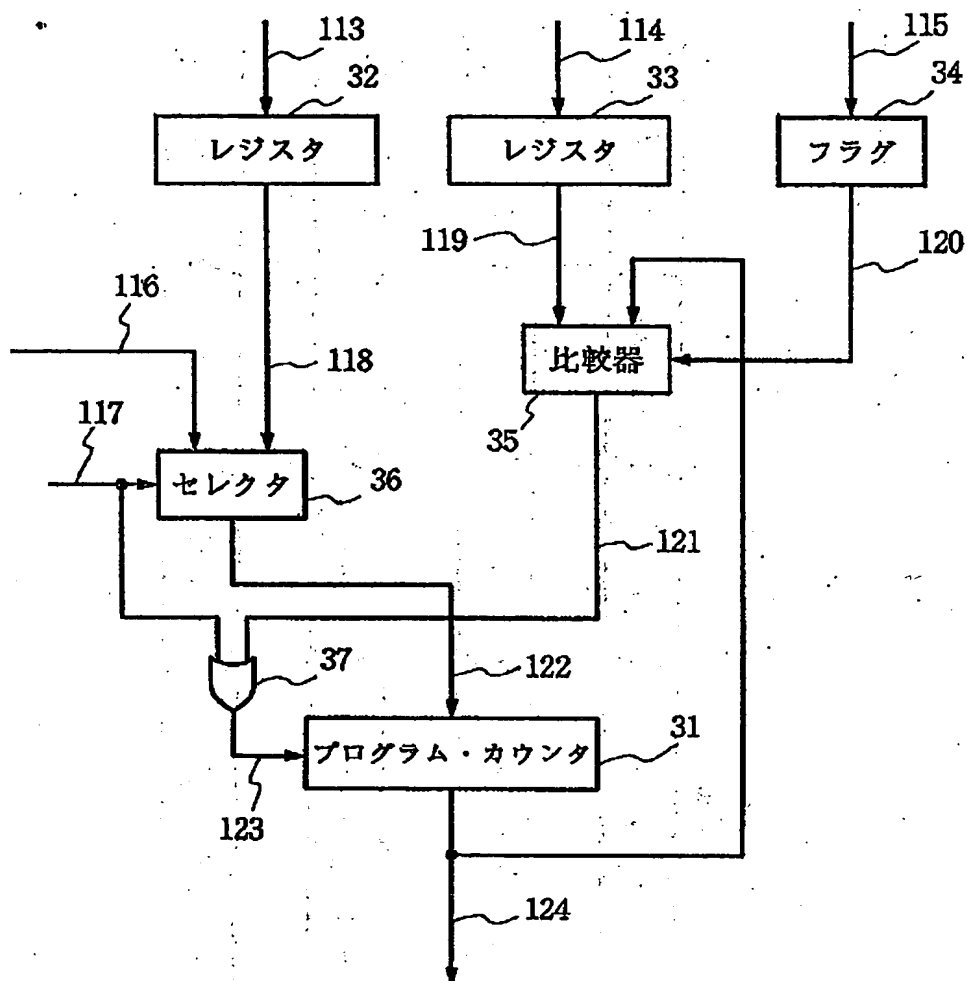
(7)

【図2】



(8)

【図3】



(9)

【図5】

